REMARKS

Applicants have reviewed the Office Action dated December 21, 2007, and the references cited therein. Claims 1-19 presented in Applicants' previously submitted preliminary amendment were all rejected. Applicants have amended the abstract to meet length limitations. Claim 19 has been amended to address the subject-matter (Section 101) rejection.

Applicants have also amended claim 1 (from which all the presently pending claims depend) to recite the additional step of second converting the abstract syntax trees back to source code. This aspect of the disclosed/described method for partitioning a source code specification enables editing of the previously partitioned program code specification. Other amendments are intended to address clarity and the specific typographical errors identified in the Office Action's "Claim Objections".

Applicants traverse the rejections of the claims over Aubury. For the reasons set forth herein below, Applicants request favorable reconsideration of the Office Action's grounds for rejecting claims 1-19 in view of Applicants' further clarifying amendments.

Please charge any fee deficiencies to Deposit Account No. 12-1216.

Summary of the Grounds for the Prior Art Rejections

1. Claims 1-19 are rejected as anticipated under 35 U.S.C. §102(e) in view of Aubury, U.S. Pat. App. Pub. 2003/0140337 (Aubury).

Detailed Remarks Responding to the Prior Art Rejections

Applicants traverse the rejection of **claims 1-19** as anticipated by Aubury because Aubury neither discloses nor suggests the claimed step of converting partitioned abstract syntax trees back to source code. The presently claimed invention is directed to a method for partitioning a source code specification for a target system (e.g., a multiprocessor system including two or more types of processors). Applicants' description of exemplary embodiments of their invention (see, page 2, paragraph 2) discloses that for a target system including a general purpose processor and a co-processor, at some point a decision is made with regard to which functions are implemented on the general purpose processor and which functions are performed on the co-processor. Thus, the specification is split into multiple partitions. Thereafter, the partitioned abstract syntax trees are converted back to source code to facilitate manual review and editing of the underlying instructions/data structures of the source code.

The second partitioning step, wherein the first and second sets of abstract syntax trees are converted back to source code, facilitates manually editing the partitioned source code specification, and thus provides the opportunity for a user to tune the program instructions of the previously generated/partitioned first and second abstract syntax tree sets. Additionally, the back-converted source code can be further processed by software compilation and hardware synthesis tools differing from the tool that performed the recited "partitioning" step. The above features of the claimed "second converting" step are also applicable to co-design of a system including a control processor and one or more control processors (see, claims 7-18). In the case of co-design of a system, the use of dedicated tools for designing/tuning the source code partitions allocated to the co-processors facilitates an optimal design for the target system.

With regard to the state of the art in this field, WO 00/38087 (identified in Applicants' background) describes a co-design system for making an electronic circuit. The system

described in WO 00/38087 includes both co-processor and software-controlled resources. This system receives a behavioral description of the target electronic system and automatically partitions the required functionality between the hardware and the software. The partitioner generates a *control/data-flow graph* from a *single* abstract syntax tree. The partitioner thereafter operates on parts of the description that have not already been assigned to resources by the user. In Aubury, once the specification has been partitioned, the resulting program structures cannot be converted back to source code to facilitate editing by a user. Aubury, which was filed by the same inventor as the above-summarized WO application/publication, does not disclose any further functionality that could be construed to correspond to Applicants' "second converting" step recited in amended claim 1.

With regard to the Office Action's specific reasons for rejecting **claim 1**, Aubury discloses (see, FIG. 4 and par. [0122] of Aubury) that a specification is converted into "the abstract syntax tree" – NOT Applicants' claimed "plurality" of abstract syntax trees – that is supplied to the partitioner. Thus, the claimed "first converting step" is not disclosed in Aubury which generates only a *single* abstract syntax tree.

Regarding the Office Action's assertion that Aubury discloses Applicants' recited "partitioning the plurality of abstract syntax trees into at least a first set and a second set of abstract syntax trees", paragraph [0125] of Aubury, to which the Office Action refers, states "The partitioner generates a control/data-flow graph (CDFG) from the abstract syntax tree." This portion of Aubury expressly teaches away from the claimed invention in that a type of data structure (a "CDFG") is generated by Aubury's partitioner that differs from Applicants' claimed *first and second sets of abstract syntax trees*. Paragraph [0125] of Aubury neither discloses nor even remotely suggests that the output of its "partitioning" step is Applicants' recited first and second "sets of abstract syntax trees." In the event the rejection of claim 1 is not withdrawn, Applicants expressly request identification of the teachings in Aubury corresponding to Applicants' claimed "first set and second set of abstract syntax trees" that are generated by during the recited partitioning step.

There is no teaching in Aubury that even remotely suggests Applicants' claimed "second converting" step. The recited "second converting" step is derived from the presently rejected claim 4. According to the now-amended claim 1, during the second converting step,

the first and second abstract syntax trees are converted back to source code. The second conversion enables editing of the previously partitioned source code. See, Applicants' published application at paragraphs [0043 and 0044] describing elements depicted in FIG. 3.

The Office Action asserts (in the rejection of claim 4 from which the amendments to claim 1 are derived) that the claimed "second" conversion back to source code is disclosed in paragraphs 143, 153 and 163. Paragraph 143 of Aubury discloses compiling the descriptions into an "RTL description" and "machine code" – neither of which is source code. Similarly, paragraph 153 describes RTL description as the generated output. A hardware description language such as Handel-C or VHKL is used as a tool, but is not the *output* of the conversion procedure disclosed in paragraph 153 of Aubury. Finally, paragraph 163 discloses "machine code" output. Thus, Aubury clearly does not disclose the "second converting" step recited in Aubury. In the event the rejection of claim 1 (and claim 4) is not withdrawn, Applicants request identification within Aubury of the claimed "source code" rendered by the recited "second converting" step.

For the reasons set forth herein above, Applicants traverse the anticipation rejection of each of the dependent **claims 2-19** that include all the recited elements of independent claim 1.

Furthermore, Applicants specifically traverse the rejection of dependent **claim 5** which recites "out-lining at least one abstract syntax tree based on profile data." The Office Action states that Aubury discloses a "profiling tool for generating a profile of the model." However, Aubury does not disclose that the profiling tool is used for "out-lining at least one abstract syntax tree based on profile data." See, pars. 13 and 39 of Applicants' published application.

Applicants furthermore specifically traverse the rejection of dependent **claim 6** which recites "out-lining at least one abstract syntax tree based on programmer provided information." The Office Action does not identify (nor could it) any teaching within Aubury with regard to using programmer-provided information to guide the claimed "out-lining" of one of the abstract syntax tree. In the event that the rejections of claims 5 and 6 are not withdrawn, Applicants request identification of the structure which is regarded as an "out-line" of an abstract syntax tree. See, e.g., par. 44 of Applicants' published application.

Conclusion

Applicants respectfully submit that the patent application is in condition for allowance. If, in the opinion of the Examiner, a telephone conference would expedite the prosecution of the subject application, the Examiner is invited to call the undersigned attorney.

Respectfully submitted,

Mark Joy, Reg. No. 35,562
LEYDIG, VOIT & MAYER, LTD.
Two Prudential Plaza, Suite 4900
180 North Stetson Avenue
Chicago, Illinois 60601-6731
(312) 616-5600 (telephone)
(312) 616-5700 (facsimile)

Date: March 21, 2008